Computational Statistics – Midterm II
Jon Braswell

Problem 1

Problem Statement

This distribution is called Extreme Value distribution (EV(a,b)) with the location parameter *a* and the scale parameter *b*. This distribution has been used widely to model extreme events, for example, annual maximum rainfall. We are interested in generating random sample from the density with *a* = 2 and *b* = 1.

Part A

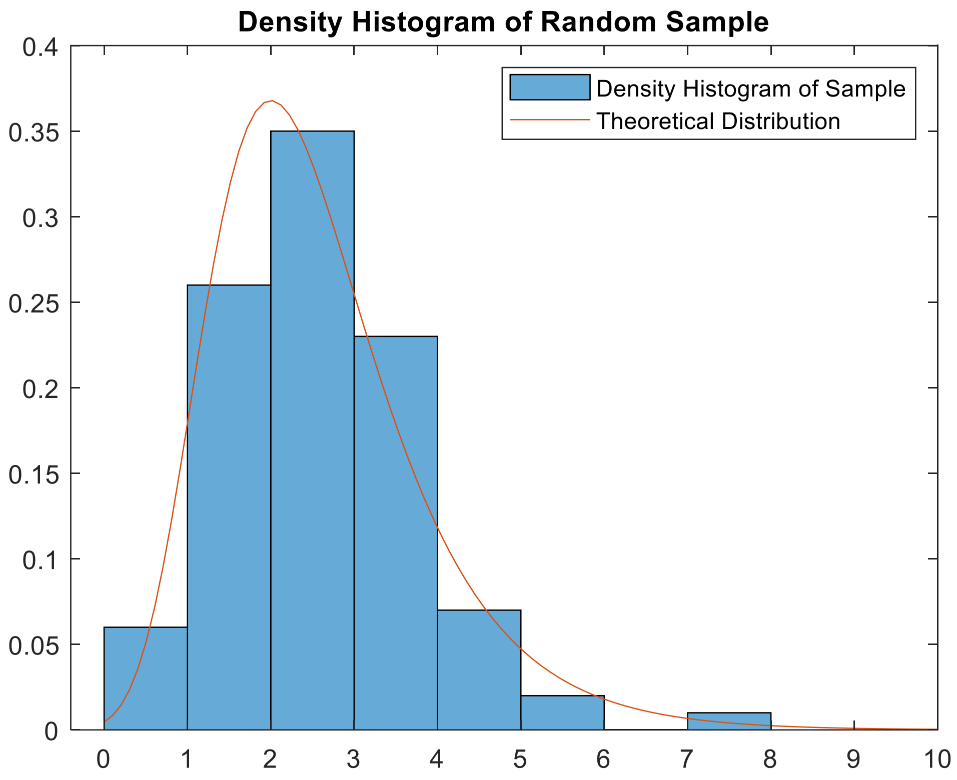
(a) Use any method of your choice to generate a random sample of size 100 from the density. Plot the density histogram of the random sample with the true density superimposed. Also provide the empirical and the theoretical CDF and compare.

Code

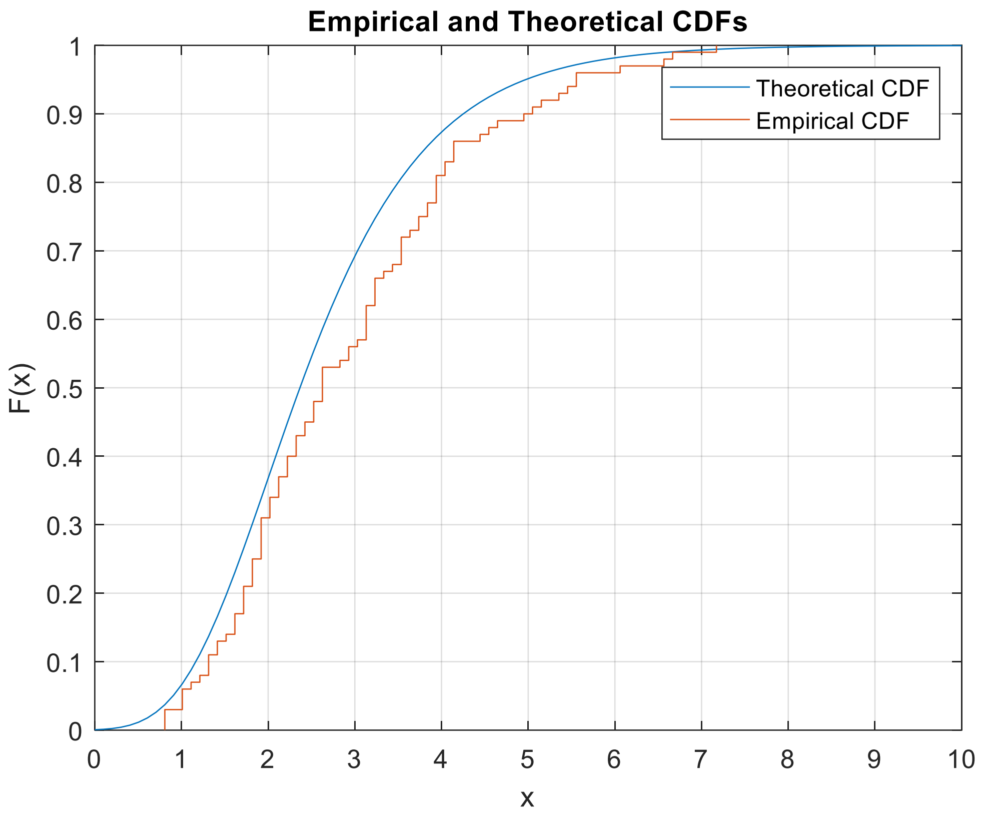
```
clear all;close all;


function [out] = extreme_cdf(in)
    out = integral(@(x)(exp(-(x-2)/1)*exp(-exp(-(x-2)/1))),...
        -100,in, 'ArrayValued',true);
end



n = 100;
step=.1;, range=[0,10];
i=(range(2)-range(1))/step;
xs=linspace(range(1),range(2),i);
ys=arrayfun(@(a) exp(-(a-2)/1)*exp(-exp(-(a-2)/1)),xs);


samp = zeros(1,n);
cum_dens = arrayfun(@(a) extreme_cdf(a),xs);
%


for i = 1:n
    u = rand(1);
    [k,dist] = dsearchn(cum_dens',u);
    samp(i) = xs(k);
end

figure(1)
histogram(samp, 'Normalization', 'probability')
hold on
plot(xs,ys)
title('Density Histogram of Random Sample')
```

```
figure(2)
plot(xs,cum_dens)
hold on
cdfplot(samp)
hold off
legend('Theoretical CDF', 'Empirical CDF')
title('Empirical and Theoretical CDFs')
```

Output



Computational Statistics – Midterm II
Jon Braswell



\

Discussion

The method chosen was the alternative CDF method whereby a standard uniform random variables was selected and was compared to the CDF results of the theoretical distribution. Whichever number most closely matched that probability was chosen and added to the sample. This was repeated until we had 100 samples.

To determine how well the data fit the theoretical distribution a density histogram was plotted alongside the theoretical distribution. We see the that the two closely match it shape. Additionally, the theoretical CDF and the empirical CDF were compared and the shapes were sufficiently similar.

Part B

(b) Using the random sample in (a), construct a MC for the location parameter a assuming the scale parameter b is known as 1. Plot the MC and the density histogram after burn-in. Give the mean, standard deviation, and 95% percentile interval for a. Discuss your findings.

Code

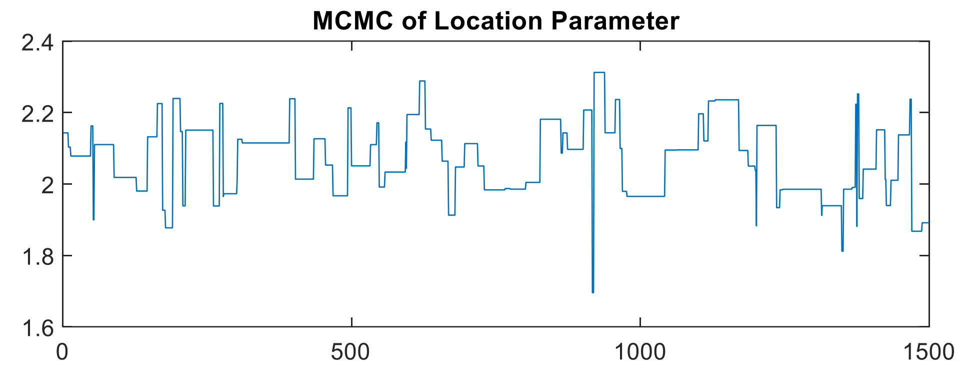
```
strg = 'prod(exp((a-x)-exp(a-x)))';
L = inline(strg,'x','a');
```

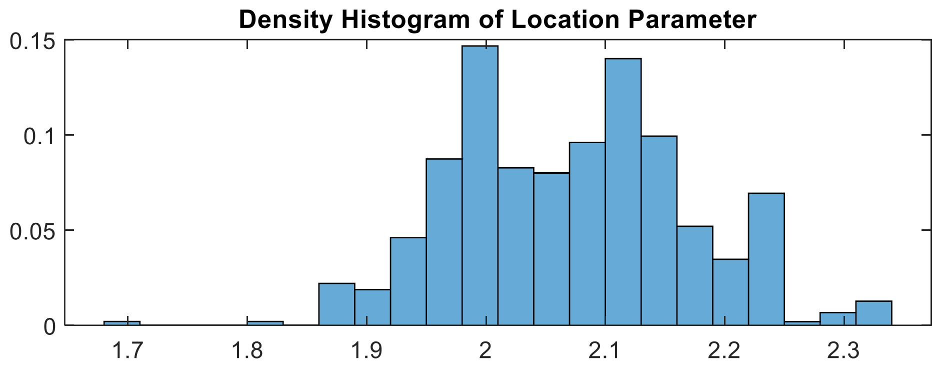
Computational Statistics – Midterm II
Jon Braswell

```
n = 2000;
theta1 = zeros(1,n);
theta1(1) = rand();
y = samp;
rate=0;
for i = 2:n
        % Generate variate from proposal distribution.
        v = exprnd(theta1(i-1));
        % Generate variate from uniform.
        u = rand(1);


        % Calculate alpha.
        alpha = min([1,L(y,v)/L(y,theta1(i-1))]);

        if u <= alpha
            % Then set the chain to the y.
            theta1(i) = v; rate=rate+1;
        else
            theta1(i) = theta1(i-1);
        end
end

n1=.25*n;
theta1=theta1(n1+1:n);
figure(3)
subplot(2,1,1)
plot(theta1)
title('MCMC of Location Parameter')
subplot(2,1,2)
histogram(theta1, 'Normalization', 'probability')
title('Density Histogram of Location Parameter');


mean = mean(theta1)
stdev = std(theta1)
lower = csquantiles(theta1,.025)
upper = csquantiles(theta1,.975)


t = table; t.mean = [mean]; t.std_dev = [stdev]; t.CI_95 = [lower upper];
disp(t)
```

Computational Statistics – Midterm II
Jon Braswell

Output





```
 mean      std_dev            CI_95             Mix_rate
______    _________    __________________    __________

2.0703    0.099248     1.881       2.2392      0.064
```

Discussion

Given a random sample from the Gumbel distribution the task was to find the location parameter using a Metropolis-Hastings sampler. Each loop of this algorithm would compare a standard uniform variate with the ratio of likelihoods (derived from the Gumbel distribution) between the old candidate for the parameter value and the new one. Candidate values were generated from an Exponential prior. If this ratio was greater than the random variate the new value was accepted for the parameter.

After generating a list of candidate parameters the average was found to be 2.07, which is very close to the true value of the parameter. The mix rate was very low, which is visible on the MCMC plot.

Computational Statistics – Midterm II
Jon Braswell

Problem 2

Problem Statement

It is known that the distribution of waiting times between events in a Poisson process with intensity λ are $Exp(\lambda)$. We would like to use this fact to generate random numbers from $Poisson(\lambda)$. Generate events $X_i = \sum_{j=1}^{i} Y_j$, $Y_j \sim Exp(\lambda)$ and then take $Z = \#\{X_i : X_i \in [l-1, l)\}, l \geq 1$, as $Poisson(\lambda)$ pseudo-rv's. Write a function to generate N such random numbers. Using your function generate N=1000 such random numbers with $\lambda = 1$. Count the outcomes in the calagories 0, 1, 2, ..., 10, 10+. Provide the density histogram and compare with the theoretical density. Discuss your findings.

(a) Describe how you might do this using a resampling method

```
function [] = resampling_method(lamb,N,L)

x = zeros(1000,1);
for j = 1:1000
    for i = 1:1000
        x(i) = sum((1/lamb)*-log(rand(N,1)));
    end
z(j) = sum(x> L & x <=L+1);
end
figure(1)
histogram(x, 'Normalization', 'probability')
title('Erlang(100,1) Density Histogram')


figure(2)
histogram(z, 'Normalization', 'probability')
hold on
histogram(poissrnd(1,1,1000), 'Normalization', 'probability')
hold off
legend('Z RV','Poisson RV with Rate 1')
title('Resampling Method')

end

resampling_method(1,100,74)
```

Discussion

Y is the time between the occurrence of **Event A** and is exponentially distributed with rate = 1.

Computational Statistics – Midterm II
Jon Braswell

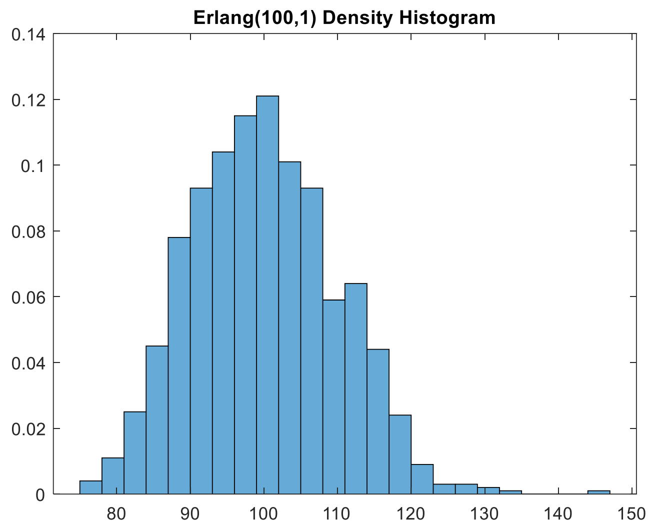
$$X = \sum_{i=1}^{N} Y_i$$

This sum is the time it takes for **Event A** to occur **N** times. This quantity has an Erlang(N,λ) Distribution.

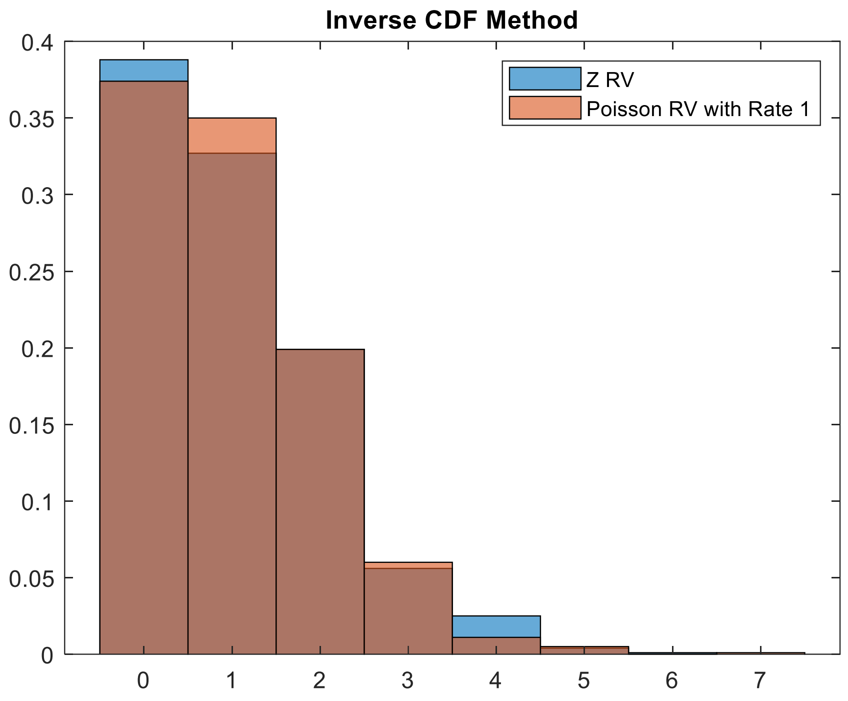
When Event A has occurred N times we say Event B has occurred.

Z will be how many times event B occurs between [L,L+1) time – which is a Poisson random variable modeling how often a rare event happens in a span of time.

The resampling method starts by using the inverse CDF method to generate N exponential random variables, then taking their sum. This is done 1000 times to generate 1000 such random variables. Plotting these variables shows we have an Erlang(N,1) distribution.



We count how many times this value falls between a number L and L+1. This gives us Z. We then run this procedure 1000 times and store our Z values. Plotting these alongside some randomly generated Poisson data shows that we have generated a Poisson random variable.



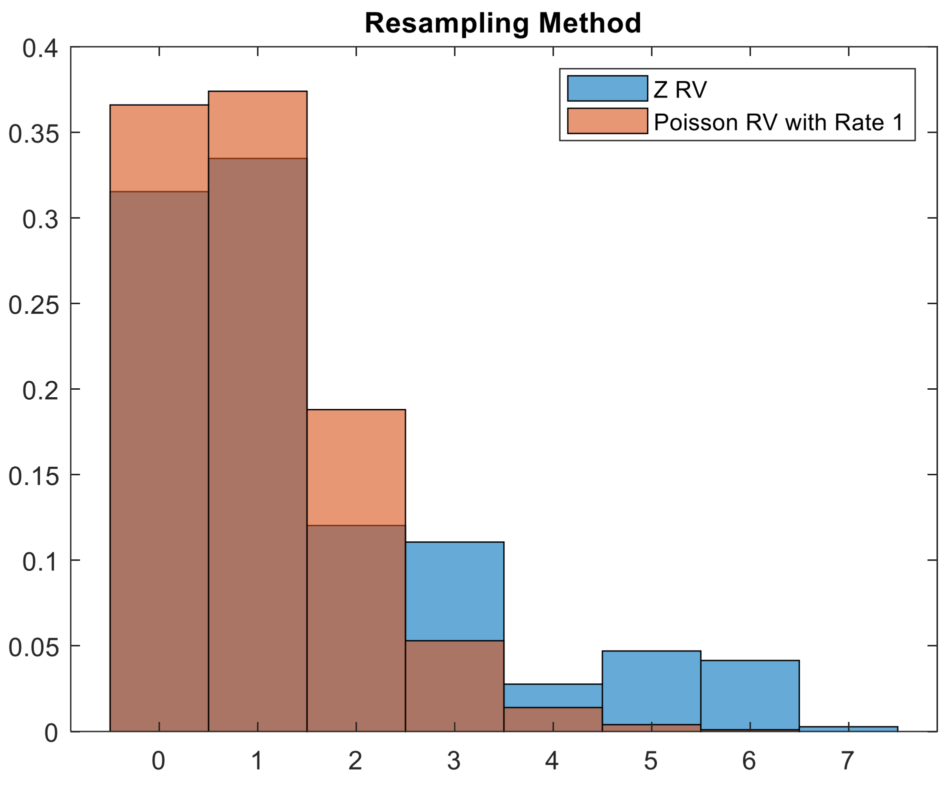
(b) Describe how you might do this using a kernel density estimation method

Code

```
% 3C
data = zeros(100,1);
for i = 1:100
    x = bootstrp(100,@sum,exprnd(1,100,1));
    data(i) = sum(x>79 & x<=80);
end

hn = 1.06*n^(-1/5)*std(data);
N = 1000;
xs = zeros(1,N);
idx = datasample(data,N);
for i = 1:N
    xs(i) = normrnd(idx(i),hn);
end
xs=round(xs(xs>=0));
histogram(xs, 'Normalization', 'probability')
```

```
figure(3)
histogram(xs, 'Normalization', 'probability')
hold on
histogram(poissrnd(1,1,1000), 'Normalization', 'probability')
hold off
legend('Z RV','Poisson RV with Rate 1')
title('Resampling Method')
```

Output



Discussion

In this case the x was formed by doing 100 bootstrap samples of exponential data with sum as the function. Z was calculated the same was as before. Doing a Normal kernel approximation took each point of Z and treated it as the center of it's own normal distribution. Then data was drawn from that distribution to recreate each point. The two do not look very similar, calling into question whether this method made any sense to begin with.

Computational Statistics – Midterm II
Jon Braswell

Problem 3

Problem Statement

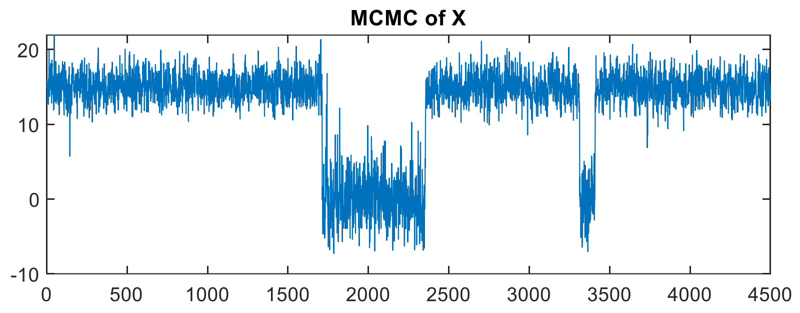
Use Gibbs sampler to generate bivariate MC, $\begin{bmatrix} X \\ Y \end{bmatrix}$ of size 5000 from a bivariate mixture normal with $\mu = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$, $v = \begin{bmatrix} 15 \\ 10 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, and $p = 0.3$.
After burn-in 10% plot each MC and histogram. Give the scatterplot of X vs Y.

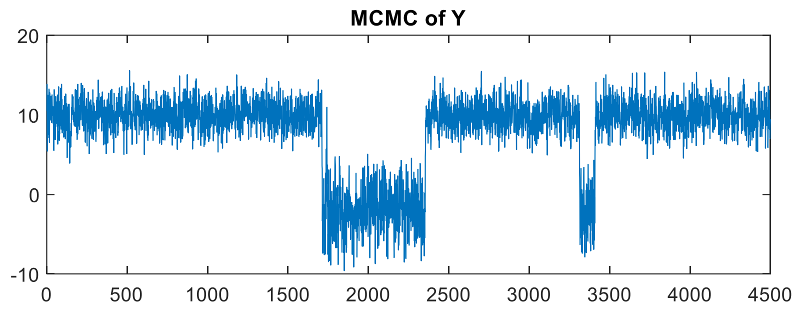
Code

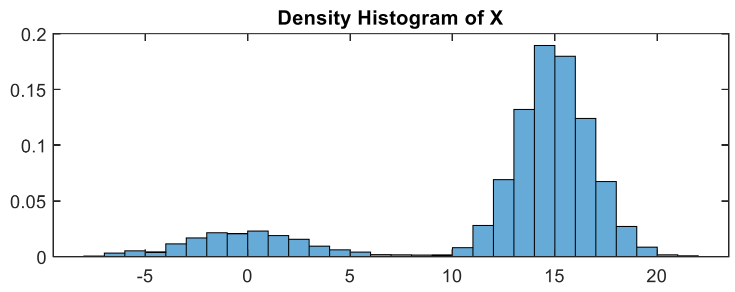
```
u1 = 0; u2 = -2; v1 = 15; v2 = 10;
a1 = 3; c1 = 1; b1 = 3;
a2 = 2; c2 = 1; b2 = 2;
mu1 = [u1; u2]; mu2 = [v1; v2];

sig1 = [a1 c1 ; c1 b1]; sig2 = [a2 c2 ; c2 b2];

p = 0.3;
n = 5000;

xgibbs = zeros(n,2);

if rand(1) < .3
    mu = mu1; sig = sig1;
else
    mu = mu2; sig = sig2;
end

vars = [sig(1,1);sig(2,2)];
stv=sqrt(vars);
rho = sig(1,2)/(prod(stv));

rho = sig(1,2)/(sqrt(sig(1,1))*sqrt(sig(2,2)));
xgibbs(1,1)=normrnd(mu(1),stv(1));
xgibbs(1,2)=normrnd(mu(2)+rho*stv(2)/stv(1),stv(2)*sqrt(1-rho^2));

for i = 2:n
    x = xgibbs(i-1,1);
    y = xgibbs(i-1,2);
     wx = p*(a1)^(-1/2)*exp(-(x-u1)^2/(2*a1))/(p*(a1)^(-1/2)...
         *exp(-(x-u1)^2/(2*a1))+(1 -p)*(a2)^(-1/2)*exp(-(x-v1)^2/(2*a2)));
    wy = p*(b1)^(-1/2)*exp(-(y-u2)^2/(2*b1))/(p*(b1)^(-1/2)...
        *exp(-(y-u2)^2/(2*b1))+(1-p)*(b2)^(-1/2)*exp(-(y-v2)^2/(2*b2)));
    xgibbs(i,1) = wy*normrnd(u1 + (y-u2)*c1/b1, det(sig1)/b1)...
        +(1-wy)*normrnd(v1+(y-v2)*c2/b2, det(sig2)/b2);
    x = xgibbs(i,1);
    xgibbs(i,2) = wx*normrnd(u2 + (x-u1)*c1/a1, det(sig1)/a1)...
        +(1-wx)*normrnd(v2+(x-v1)*c2/a2, det(sig2)/a2);
end

burn=n*.1;
nMC=n-burn;
MCx=xgibbs(burn+1:n,1);
MCy=xgibbs(burn+1:n,2);


figure(1)
subplot(2,1,1)
plot(MCx)
title('MCMC of X')
subplot(2,1,2)
plot(MCy)
title('MCMC of Y')

figure(2)
subplot(2,1,1)
histogram(MCx, 'Normalization','probability')
title('Density Histogram of X')
subplot(2,1,2)
histogram(MCy, 'Normalization','probability')
title('Density Histogram of Y')

figure(3)
scatter(MCx,MCy)
xlabel('X');ylabel('Y')
title('Scatter Plot of X and Y')
```

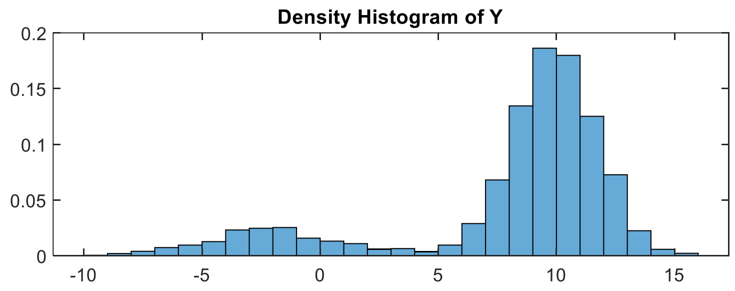
Computational Statistics – Midterm II
Jon Braswell

Output

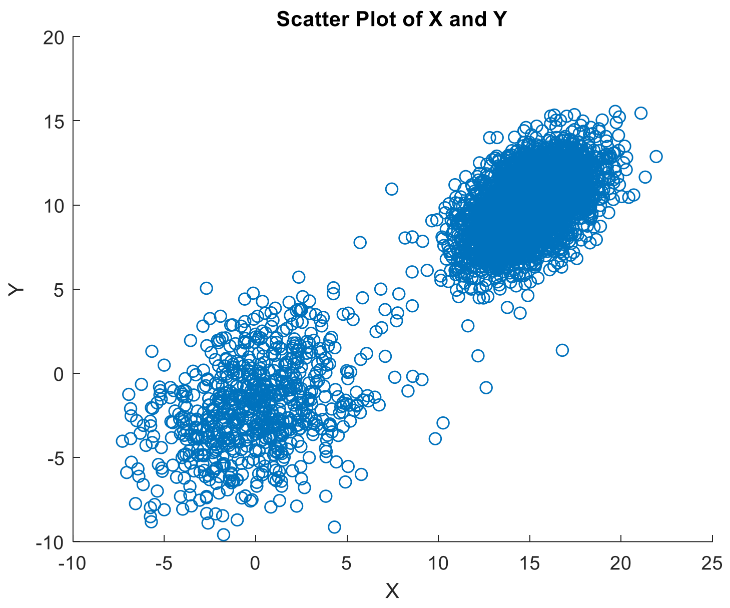








Computational Statistics – Midterm II
Jon Braswell



Discussion

The aim was to generate bivariate MC of size 5000 from a bivariate mixture model with the parameters provided. The marginal distributions were known ahead of time, allowing us to use them explicitly with the parameters provided. The data was successfully generated, as is shown in the plots.

In Figure 1 we see the MCMC plotted for both X and Y. For both values we see the chain hovering around the two different population means, spending more time in the higher values provided by the second part of the mixture.

In Figure 2 we see a density histogram of X and Y, which reinforced our impressions from Figure 1. The centers of mass are in the areas of the provided means of the two populations we were trying to generate data on.

Figure 3 shows this data in another way, showing two clumps of data- one from each population.